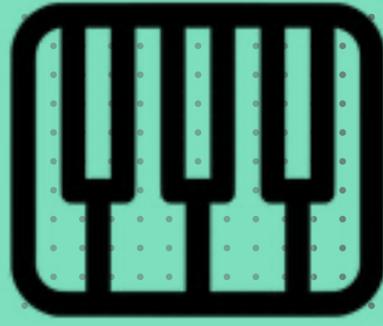


Itsp



INTELLIGENT TUTORING SYSTEM OF PIANO

INDEPENDENT STUDY OF NCCU CS

MONOGRAPHER: KUEI-YU TSAI
THESIS ADVISOR: DR. TZU-CHIEH TSAI

INTRODUCTION

智慧輔導系統（ITS）最早的概念是自1970年代起的電腦輔助教學（CAI），初期的ITS系統是為能教導、執行特定項目並動態適應各式情況。到了現代的ITS系統嘗試複製老師或助教的角色並執行自動化教學，如提出問題、選擇問題和提供回饋。

我所嘗試建立的鋼琴智慧輔導系統（ITSP）其注重的重點是互動式學習，所選用之方式為一連串的鋼琴遊戲挑戰。於此構想中，其所需要的將不如以往所見的ITS那般複雜，而是摒除現在ITS如教授那般所做的智慧型授課，比較像是一位老師針對不同程度的學生所提供的訓練，遊戲之課程內容由人為修訂而適合學生程度之課程順序則是系統來完成，在這些互動學習包括系統評比的分數以及在學習過程中學生的行為及回饋來驗證是否對鋼琴學習提升興趣及幫助。

ALGORITHM

Elo Rating Algorithm System 是指由匈牙利裔美國物理學家Arpad Elo所創建的一個用於衡量各類對弈活動水準的評分方法，是當今事件上對於對一水準評估的公認權威標準。

而我會將課程於此演算法中作為學生的對手，每個不同難度的課程分別會有相應不同的分數（難度分數），當學生開始進行課程時也會獲得一個起始分數（程度分數）。在課程遊戲進行時會以音符順序的正確與否來獲得相應分數（課程分數），算出新的分數後根據判斷會做為程度或是難度之分數來協助分析後續課程序列。

而用於判斷課程內容分數之演算法為LCS，透過問題可以分解成更小、更簡單的子問題，其又可以分成更多的子問題，並且使用Dynamic Programming演算法來解決，這樣子問題的解就可以被儲存起來，而不用重複計算，因此整個課程就變得更加容易。

對於課程內容皆為難度不一的樂句，使用LCS演算法可協助每個不同課程去做調整並設立不同難度之通關標準。

DATA

```
<div class="screen step0">

<h3>Intellinet Tutoring System of Piano</h3>
<h4>Press any note to begin...</h4>
<p class="intro">monographer: Kuei-Yu Tsai</p>
<p class="intro">thesis advisor: Professor Tzu-Chieh Tsai</p>
</div>

<div class="screen step1">
<br>
<div class="lock-input">
<div class="note"><svg><use xlink:href="#note"></use></svg></div>
</div>
<p class="hint">Hint: </p>
</div>
```

HYPertext Markup Language Part

```
.lock-input {
background: ■#ffff1dc;
padding: 50px 75px;
border-radius: 15px;
box-shadow: 0px 2px 5px □#000;
margin: 20px 0;
text-align: center;
overflow: hidden;
transition: 150ms ease;
display: inline-block;
}
```

CASCADING STYLE SHEETS PART

```
if (navigator.requestMIDIAccess) {
  console.log('this browser supports WebMIDI!');
  navigator.requestMIDIAccess().then(onMIDI Success, onMIDI Failure);
} else {
  console.log('WebMIDI is not supported in this browser.');
  document.querySelector('step1').innerHTML = 'Error: This browser does not support WebMIDI.';
}

function onMIDI Success(midiAccess) {
  var inputs = midiAccess.inputs;
  var outputs = midiAccess.outputs;

  for (var input of midiAccess.inputs.values()) {
    input.onmidimessage = getMIDIMessage;
  }
}
```

JAVA SCRIPT PART

```
var score = dp[n1][n2] * 100 / lock2c.length;
var rate = 1 / (1 + Math.pow(10, (currentPoint - point2) / 400));
currentPoint = currentPoint + rate * score * 2;

```

SIMPLE ELO RATING APPLICATION

```
n1 = active.length;
n2 = lock2c.length;
dp = new Array(); //先宣告一堆
for(var i = 0; i <= n1; i++) { //一維長度為i, i為變數，可以根據實際情況改變
  dp[i] = new Array(); //宣告二維，每一側一維陣列裡面的一個元素都是一個陣列；
  for(var j = 0; j <= n2; j++) { //一維陣列裡面每個元素陣列可以包含的數量p，p也是一個變數
    dp[i][j] = 0; //這裡要數數初始化，我這邊初始化為0，後面在用所需的值覆蓋裡面的值
  }
}
for (var i = 1; i <= n1; i++) {
  for (var j = 1; j <= n2; j++) {
    if (active[i - 1] != lock2c[j - 1]) {
      dp[i][j] = Math.max(dp[i - 1][j], dp[i][j - 1]);
    } else {
      dp[i][j] = dp[i - 1][j - 1] + 1;
    }
  }
}
```

SIMPLE LCS(DP) APPLICATION

METHODS

使用JavaScript搭配HTML並以CSS完成，並以經典之BEYER鋼琴教本作為課程教材使用。

輸入端使用外接之MIDI設備，並用JS之Web MIDI API讓使用者可以使MIDI設備對於JS編撰之程式加以控制。程式內容中包含前述提及之Elo Rating Algorithm在npm.js中提供，並以array簡易儲存LCS中之Dynamic Programming需求。而輸出端則是由Google Chrome這款被廣泛使用並支援Web MIDI API使用之瀏覽器，以HTML架構網頁遊戲並用CSS對程式輸出畫面做美化調整。



CONCLUSION

此專題的最終目標是希望系統能像一位家教針對不同難度的學生所提供的鋼琴訓練遊戲，以及包含人為修訂的課程及輔導學生學習鋼琴過程的系統，課程以此方式進行，如同現今在市面流傳許多的鋼琴演奏遊戲，並加入更多的音樂知識教學用以精進樂器技巧，故猜測會更吸引學生來學習。

此專題之目標族群定為約是10歲以下的學童及學齡前兒童來使用，並且以自身過往有長期學習的鋼琴來做為目標，而MIDI鍵盤及電腦相對來說硬體門檻較低。透過此系統，希望能如同先前ITS在其他領域的發展，雖然專題規模簡單但也期許能對教師的教學及學生的學習產生正面的影響，同時能提升對於鋼琴學習的遊戲性及成就感。